Incorporating Scheduling In the Optimal Design of Muitiproduct Batch Plants

by

D. Birewar and I. Grossmann

EDRC-06-34-87

INCORPORATING SCHEDULING IN THE OPTIMAL DESIGN OF MULTIPRODUCT BATCH PLANTS

Deepak B. Birewar

and

Ignacio E. Grossmann*

Department of Chemical Engineering
Carnegie-Mellon University
Pittsburgh, PA 15213
October 1987

Prepared for presentation at Annual AIChE Meeting, New York, November 15-20, 1987
Session : Computer Aided Design of Batch Processes

UNPUBLISHED

•**Author to whom correspondence should be addressed**

Abstract

Anticipating the impact of scheduling at the design stage can result in significant savings in the capital cost of multiproduct batch plants. In this paper this idea is applied to multiproduct batch plants with one unit per processing stage. Efficient scheduling models are developed for minimization of cycle time for Unlimited Intermediate Storage and Zero Wait policies in mixed product campaigns. Based on these models, it is shown that simplified constraints that account for these scheduling policies can be incorporated in the optimal design problem. Numerical examples are presented for both scheduling and design problems.

Introduction

Incorporating constraints that account for scheduling effects in the optimal design of multiproduct batch plants is a very difficult task. Thus, previous procedures for design of these plants have assumed only the simplest type of scheduling policy, namely campaigns of single products with no intermediate storage. In this policy it is assumed that the products will be produced in a sequence of campaigns each devoted to a single product. In this way the scheduling is characterized through cycle times for each product which are then included into a single time constraint for the optimal design problem. This type of assumption which greatly simplifies the design problem, is the one that has been used in previous methods (e.g. Sparrow et al , [1975], Grossmann and Sargent, [1979], Flatz, [1981], Yeh and Reklaitis, [1985]).

The important limitation in these methods, however, is that the simplified scheduling policy that is assumed at the design stage can greatly overestimate the time requirements. Therefore, this can have the effect of producing large overdesigns for the equipment when a fixed time horizon is considered for specified production requirement. It would be clearly desirable to consider at the design stage more efficient scheduling policies such as mixed product campaigns, where sequencing of batches of different products can reduce idle times to increase the utilization of equipment. It is precisely the purpose of this paper to accomplish this objective. As an initial step, the case of multiproduct batch plants with one unit per stage will be considered.

The problem that is specifically addressed in this paper is as follows. Given is a plant with M batch stages with one unit per stage. Production rates for a given time horizon, processing times and size factors for N_p products that are to be manufactured are also given. It is assumed that the plant is of the multiproduct type [Rippin,1983] in which all the products follow the same processing route through the M stages. The problem then consists in determining sizes of batch equipment, the production scheduling, and the possible use of intermediate storage that will minimize the investment cost of the plant.

For the scheduling aspect, mixed product campaigns will be considered for the two following cases: Unlimited Intermediate Storage (UIS) and Zero Wait (ZW) policy. As is well known [Ku and Karimi, 1986], these correspond to limiting cases with the former being the most efficient, and the latter the most conservative scheduling policy. The proposed strategy will consist of developing for these two policies simplified expressions for time constraints that can be readily incorporated into the design problem. The detailed scheduling is obtained at a second stage, where the question of intermediate storage can be addressed by analyzing the two limiting policies.

In order to handle the scheduling problem effectively, new formulations are proposed for the UIS case, and the ZW policy, and where the objective that is considered is the minimization of cycle time. The former policy involves an MILP problem whose optimal objective function value can be obtained by a simple analytical expression. The latter policy involves the solution of a 0-1 minimax assignment problem that is based on time slacks for any two consecutive batches, and which can be solved as an LP in most cases. Both of these formulations can be solved with modest computational effort even for a large number of product-batches. Using as a basis these formulations, the derivation of simplified constraints for design are then presented. It will be shown that these constraints can be incorporated into the NLP model that has been reported previously in the literature for multiproduct batch plants.

Examples are presented first to illustrate the efficiency of the proposed formulations for the scheduling of the UIS and the ZW policies. Examples are then also presented to illustrate how these two scheduling policies can be incorporated at the design stage with the proposed simplified constraints.

Comparison of Sequencing Policies

In order to illustrate the difference of scheduling a multiproduct batch plant with single product campaigns as opposed to using mixed product campaigns, consider example 1 in Table 1 involving 3 products and 3 stages, where 3 batches of each product must be manufactured. It is assumed that set-up and clean-up times are negligible.

If we do not allow for intermediate storage, the lowest total time (makespan) that is required for the single product campaign is 48 hrs as shown in Fig. 1(a). Note that the cycle time of products A, B, C are 5 , 4 and 5 hrs, respectively. To reduce the makespan, we could consider *mixtd product campaign** where the three products A, B, C, are produced in 3 cycles. The most conservative policy would be the zero-wait (ZW) policy where a batch upon completion in a stage must be transferred immediately to the following stage. The shortest makespan we can obtain is of 42 hrs by using the sequence C-A-B as seen in Fig. 1(b). Note that in this schedule the cycle time for a single sequence of C-A-B is 13 hrs. Also note that stage 1, which limits the cycle time, involves slack times of 4 hours between products A and B. To further reduce the makespan, we could consider the use of intermediate storage in a mixed campaign. In particular, with the unlimited intermediate storage policy (UIS), the shortest makespan that we obtain is of only 38 hrs with the sequence C-A-B as seen in Fig. 1(c). Note that in this case the cycle time of a single sequence C-A-B is reduced to 11 hours and that stage 3 limits this cycle time.

Thus, from the above example we can conclude that by sequencing the batches of different products in a mixed campaign, we can decrease significantly the makespan when compared to using single product campaigns with no intermediate storage. Furthermore, the UIS and the ZW policies are limiting cases for sequencing, with the former providing the shortest makespan, and the latter with the longest. For the design problem, where a fixed horizon time is considered (typically one year), it then follows that the UIS policy requires the smallest equipment sizes at each stage, while the single product campaigns requires the largest. The ZW policy will in general require equipment sizes that lie in between.

In order to derive appropriate time constraints for sequencing in the optimal sizing problem, it is useful first to present general MILP formulations for minimizing the makespan with the unlimited storage and zero-wait policies. It will be shown that the difficulty involved in solving these formulations can be circumvented by replacing the minimization of the makespan by the minimization of the cycle time. This objective, which is suitable for *long Kangt hohJLzonb,* will then be used as a basis for deriving the simplified time constraints for the design problem.

MILP Models for Minimizing Makespan

The problem of determining the product sequencing that leads to the shortest makespan for the UIS and ZW policies will be formulated as MILP problems in this section. In order to derive the corresponding models, N_c identical cycles will be assumed for the production. Each cycle consists of N batches involving N_p products. As an example consider Table 2 where the production task consists of manufacturing 6 batches [N=6] of products A, B, C, D, E and F [$N_p = 6$] in five cycles [$N_c = 5$]. These product-batches are to be manufactured in a plant with 4 processing stages.

Since we do not know apriori the sequencing of the N product-batches in each cycle, we will consider a sequence of N production slots where the assignment of each batch to each of these slots must be determined. The potential assignment of a batch i, $i=1,...N$, to a production slot ℓ, $\ell=1,...N$, will be denoted by the 0-1 variable $Y_{i\ell}$, where a value of one implies that batch i is assigned to production-slot ℓ. The following constraints must be satisfied for these 0-1 variables:

Every slot ℓ must be assigned exactly to one product-batch i:

$$\sum_{i=1}^{N} Y_{i\ell} = 1 \qquad \ell = 1,...,N \tag{1}$$

Every product-batch i must be assigned to exactly one slot ℓ:

$$\sum_{\ell=1}^{N} Y_{i\ell} = 1 \qquad i = 1,...,N \tag{2}$$

As seen in Fig. 2, the following variables are required to model the times associated at each stage j, $j = 1,...M$, of the production slots ℓ, $\ell = 1,...N$:

$TF_{j\ell}$ = time at which processing in time-slot ℓ of stage j is finished.

$TI_{j\ell}$ = time at which processing in time-slot ℓ of stage j is started.

Given the fixed processing times t_{ij} of batch i in stage j, the two above variables are related in terms of the assignment variables $Y_{i\ell}$, by the equation:

$$TF_{j\ell} = TI_{j\ell} + \sum_{i=1}^{N} \tau_{ij}\, y_{i\ell} \qquad \ell = 1, \ldots, N \; ; \; j = 1, \ldots, M \qquad (3)$$

where $TI_{11} = 0$.

From Fig. 2, it is clear that the start time of slot $\ell+1$ at every stage j requires that the processing of slot ℓ be finished. That is,

$$TI_{j\ell+1} \geq TF_{j\ell} \qquad \ell = 1, \ldots, N-1 \; ; \; j = 1, \ldots, M \qquad (4)$$

As for the completion and start time relations for two successive stages, these depend on the sequencing policy that is used:

a) For the UIS policy where there is the possibility of storing the batch produced in stage j, the start time of stage $j+1$ can be performed any time after the completion of stage j:

$$TF_{j\ell} \leq TI_{j+1\,\ell} \qquad \ell = 1, \ldots, N \; ; \; j = 1, \ldots, M-1 \qquad (5a)$$

b) For the ZW policy where no intermediate storage is available, and no idle times are allowed for the processing between stages, the start time of stage $j+1$ has to coincide exactly with the completion time of stage j:

$$TF_{j\ell} = TI_{j+1\,\ell} \qquad \ell = 1, \ldots, N \; ; \; j = 1, \ldots, M-1 \qquad (5b)$$

In order to define the total time (makespan), it is convenient to define first the cycle time associated with each stage j:

$$CT_j = TF_{jN} - TI_{j1} \qquad j = 1, \ldots, M \qquad (6)$$

The total time must then be greater or equal to the total processing time required for each stage. Since N_c cycles are considered, this leads to the inequality.

$$T_t \geq TI_{j1} + N_c\, CT_j + [TF_{MN} - TF_{jN}] \qquad j = 1, \ldots, M \qquad (7)$$

where the first and third term in the right hand side, correspond to head and tail times, respectively (for example, see Fig. 3).

Finally, the minimization of the makespan implies the objective function,

$$min \quad T_x \tag{8}$$

This objective function subject to the constraints (1M7) defines problems MILP1 and MILP2 with which one can determine the optimal sequencing for the UIS and ZW policies, respectively. Note that MILP1 involves constraint (5a) while MILP2 involves constraint (5b).

In order to provide some insight into the computational requirements of these MILP models consider the example given in Table 2. For the UIS case the optimal sequence is (E-A-B-F-D-C) with a total makespan of 427 hrs and a cycle time of 80 hrs, as shown in Fig. 4(a). The predicted optimal sequence for the ZW policy is (E-D-B-A-F-C) as shown in Fig. 4(b), with a total makespan of 505 hrs and a cycle time of 97 hrs. In Table 3 the problem sizes and computational statistics for minimization of makespan are presented. Note that both, MILP1 and MILP2 have a non-zero gap with respect to the LP solution where the integrality constraints on the 0-1 variables are relaxed. Hence, both problems require a relatively substantial effort in the branch and bound procedure.

Since solving problems MILP1 and MILP2 can clearly become computationally expensive for larger problems, it is convenient to exploit some of the features of the design problem. In particular since the horizon time H is usually long (e. g. operating time for one year), the number of cycles N_c that is necessary for total production will be typically rather large. This would suggest that an alternative criterion for the scheduling could be the minimization of cycle times since the contribution of heads and tails for the total time as given by constraint (7), will be negligible for a large number of cycles N_c. In the next two sections it will be shown that considerable simplifications are possible by considering the minimization of cycle times (instead of total makespan) of the UIS and ZW policies.

Minimization of Cycle Time for UIS Policy

Consider equation (6) which defines the cycle time for each stage. If we add (3) over the *l* slots and substitute (1) and (2) into this equation, we obtain

$$CT_j = \sum_{i=1}^{N} \sum_{\ell=1}^{N} y_{i\ell} t_{ij} = \sum_{i=1}^{N} \Big[\sum_{\ell=1}^{N} y_{i\ell}\Big] t_{ij} = \sum_{i=1}^{N} t_{ij} \qquad (9)$$

If we neglect the first and third term of the right hand side in (7) as N_c can be assumed to be large, then the minimum cycle time is given by:

$$CT_{MIN} = \frac{T_t}{N_c} = \max_{j=1,\ldots,N} \{CT_j\} \qquad (10)$$

From (9) it then follows that,

$$CT_{MIN} = \max_{j=1,\ldots,N} \Big\{ \sum_{i=1}^{N} t_{ij} \Big\} \qquad (11)$$

This theoretical minimum for the cycle time is always attainable for the UIS case. The constraints in (4) are automatically satisfied by (11). Constraints (5a) can also always be satisfied by shifting the upstream times to the left and the downstream times to the right with respect to the bottleneck stage j^* that defines the cycle time in (11). Furthermore, what (11) implies is that for UIS the minimum cycle time is *independent* of the sequencing of batches. Thus, the derivation of a schedule for UIS with minimum cycle time reduces simply to selecting *any* sequence of product-batches, and setting slack times of bottleneck stage(s) to zero. This then also implies that if we solve the following problem:

min CT

s.t. $CT \geq CT_j \quad j = 1,\ldots,N$ *(MILP3)*

and constraints (1)-(5a) *and* (6)

it will have a zero-gap between the relaxed LP and the integer solution. With the solution of MILP3 the timings of the schedule for UIS policy can be obtained. MILP3 has been applied to the problem in Table 2 that involves production of batches for 6 different products in a plant with four production stages. As seen in Table 3, MILP3 has zero gap predicting the minimum cycle time of 80 hours which is also predicted by equation (11). As will be shown later in the paper, equation (11) can be used as a basis to derive time constraints for the optimal design problem with UIS policy.

Minimization of Cycle Time for Zero Wait Policy

In the case of the ZW policy the equation in (11) provides only a *lowtA bound* to the cycle time for the ZW policy. The reason is that the equality constraints in (5b) may not be satisfied, since for the ZW policy there is no freedom of shifting times upstream and downstream as was the case in the UIS policy. Furthermore, stages that limit the cycle time will contain in most cases non-zero slacks. This then means that if we were to solve problem MILP3, with constraint (5b) instead of (5a), there would be a non-zero gap in the MILP, which would still make it computationally expensive to solve. For this reason, a new MILP formulation for the ZW policy will be developed which exhibits zero gap and can in fact be solved as an LP problem in most instances.

The basic idea behind the new formulation is as follows. For the ZW policy the optimal cycle time will, in general, not be equal to the minimum cycle time of UIS (>c.e. CT^{1M}_{MIN}) because all the stages including the ones that define bottleneck will often exhibit idle times (*iJjickl)* between some of the consecutively produced product-batches (e.g. see Figs. 1 and 4). It should be noted that these *Alack** are only a function of consecutive pairs of batches [Wismer, 1972]. As an example consider the two batches in Fig. 5. It can be seen that the slack in stage 1 is 1 hr., in stage 2 it is zero, in stage 3 it is 2 hrs. Therefore, the slacks for each pair of batches can be easily computed a priori (see also Appendix I).

Based on the observation that in ZW policy two consecutive batches define the minimum slack time at each stage, it is possible to use this insight to develop a method for sequencing that is considerably faster that the previous ones. It is then natural to define the following binary variable for any two successive product batches *i* ¥ fe:

$$YC_{ik} = \begin{cases} 1 & \text{if product-batch } k \text{ is produced after } i \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

As every product-batch is produced exactly once in each cycle, it will appear exactly once in the first place and exactly once in the second place in the pairs *(i, k)* of product-batches that are produced during a production cycle. Therefore the following two assignment constraints must apply*,

$$\sum_{k=1}^{N} YC_{ik} = 1 \qquad i = 1, \ldots, N \tag{13}$$

$$\sum_{i=1}^{N} YC_{ik} = 1 \qquad k = 1, \ldots, N \tag{14}$$

The cycle time of any stage is composed of the batch processing times of each product-batch produced in that stage and the slacks that are forced to exist between some of the two consecutive product-batches. Since the overall cycle time CT must be greater than or equal to the cycle time associated with each stage, the following constraint must be satisfied:

$$CT \geq \sum_{i=1}^{N} \sum_{k=1}^{N} YC_{ik} SL_{ikj} + \sum_{i=1}^{N} t_{ij} \qquad j = 1, \ldots, M \tag{15}$$

where SL_{ikj} defines the slack (or forced idle time) in stage j when product-batches i and k are produced in succession, in that order. The data for the slacks SL_{ikj} can easily be generated a priori by examining every pair of products *i, k* as was explained previously. As an additional example, consider the three products and three stages in Table 1. Using the procedure described in Appendix I, examination of each pair of products yields the slack times given in Table 4.

The objective function of minimization of cycle time,

$$\min \; CT \tag{16}$$

subject to (13)-(15) defines then MILP4 which is a 0-1 minimax assignment problem.

---

*For simplicity the condition $i \neq k$ is not stated in the equations

In Appendix II it is proved that if there is only one stage that limits the cycle time, pure integer solutions in MILP4 can be obtained through the solution of the relaxed LP where the YC_{ik} variables are treated as continuous between values of zero and one. For the case of two or more stages that limit the cycle time, it is proved that the relaxed LP has zero gap with respect to the optimal MILP solution. This means that MILP4 can often be solved as an LP problem or else requires the examination of few branches in a branch and bound procedure.

The formulation MILP4 has been applied to Example 2. As seen in Table 3 the cycle time of 97 hrs was obtained by simply solving the relaxed LP. Thus, the CPU time is much smaller than in MILP2. The optimal sequence that was obtained is E-D-B-A-F-C which is same as in Fig. 4(a). It should be noted that actual implementation of this sequence can be performed by starting at any product; that is:

E-D-B-A-F-C , D-B-A-F-C-E , B-A-F-C-E-D

A-F-C-E-D-B , F-C-E-D-B-A , C-E-D-B-A-F

all have the same cycle time. The one leading to minimum makespan could be chosen by direct examination of the alternatives or with the formulation presented in Appendix III.

Another important point in the model MILP4 is that in principle it is possible to obtain a sequence with subcycles of batches. For example, given the six products (A, B, C, D, E, F) the constraints in (13) and (14) would satisfy the assignment of variables for which the sequence is given by the subcycles (A-B-C) and (D-E-F). If each of these subcycles is performed a large number of times one after the other, then the error introduced by such a solution would only be the slack times for the transition from one subcycle to the other. It should be noted however, that the occurrence of subcycles will only tend to arise in large problems. Also, subcycles could be eliminated by the introduction of subtour elimination constraints as in the travelling salesman problem, but this can greatly increase the difficulty of solution of the problem [Papadimitriou and Steiglitz,1982].

Finally, it should be noted that the concept of using the slacks in MILP4 can easily be extended to account for set-up and clean-up times. The data for minimum clean-up and set-up times when product k is produced after product i can be incorporated into slack tables so as to satisfy both ZW conditions and the clean-up and set-up times. For example, say the clean-up and set-up data in example 1 is such that when product A is produced after product B, stage 1 requires 2 hrs and stage 2 requires 1 hr. The slacks in Table 4 for this particular product combination show that stage 1 has zero slack. Thus, a slack of 2 hrs will have to be added in stage 1. To satisfy constraint (5a) all the downstream stages will have to be 'shifted forward' by two hrs. *JL.I.* fourth row of Table 4 will have to be changed from 0, 1, 4 to 2, 3, 6.

Design and Sizing of Batch Plants

Having considered the scheduling in mixed product campaigns with the UIS and ZW policies, the optimal design problem will be considered next. The formulation NLP1 for designing multiproduct batch plants with one processing unit in each stage, with single product campaigns and no intermediate storage is as follows [Grossmann and Sargent ,1979]:

$$\min \sum_{j=1}^{M} \alpha_j V_j^{\beta_j} \qquad \text{(NLP1)}$$

s.t.

$$V_j \geq S_{ij} B_i \qquad j = 1....M \; ; \; i = 1....N_p$$

$$\sum_{i=1}^{N_p} \frac{Q_i T_{L_i}}{B_i} \leq H$$

$$V_j \geq 0 \quad j=1,....,M \; ; \quad B_i \geq 0 \quad i=1,....,N_p$$

and where

$$T_{L_i} = \max_{j=1,....,M} \{ t_{ij} \} \qquad i = 1,....,N_p$$

In the above formulation the objective consists of minimizing the investment cost in terms of the volumes V_j of the units at each stage. The first constraint defines the volume requirements at each stage in terms of the size factors S_{ij} and the batch sizes B^. The second inequality is the horizon constraint for single product campaigns; namely, the sum of cycle times T multiplied by the number of batches [Qy / By] must be less than or equal to the horizon time H, where Qy are the fixed production demands for each of the products. Finally, the cycle time T_{L_i}, for each product *JL,* is equal to the maximum of the processing times tyy of that product over all the stages.

It is clear that in order to consider scheduling policies different from single product campaigns, the second constraint (horizon constraint) in NLP1 must be replaced by other suitable expressions. In the case of the UIS policy this can be accomplished as follows.

Consider the definition of the minimum cycle time in (11). By defining a cycle to extend over the horizon time H so as to include all the product-batches *V* = 1,..,N that are required to satisfy the total production demand, the total cycle time for each stage *j* is given by* ,

$$CT_j^H = \sum_{i'=1}^{N} *£,_j \qquad y = 1....M \tag{17}$$

where N is the number of product-batches in a cycle. Since the cycle extends over the whole horizon time, N represents the *total* number of product-batches that will be produced in the time horizon H. In terms of individual products, N is given by,

$$N =]JT_{i=1} \; nji \tag{18}$$

where the number of batches n-. of each product *I* is in turn given by

$$n_: - \frac{}{B_i} \qquad i \ « \ 1,...,N_p \tag{19}$$

From (17), (18) and (19) it then follows that.

* In this section JL^{01} corresponds to the index of product-batches, /C for the products.

$$CT_j = \sum_{i=1}^{N} n_i t_{ij} = \sum_{i=1}^{N} \frac{Q_i}{B_i} t_{ij} \qquad j = 1, \ldots, M \tag{20}$$

Since the horizon time H must be greater than or equal to the total cycle time of each stage j (neglecting the heads and tails), then from (20), the simplified horizon constraints for UIS policy reduce to:

$$\sum_{i=1}^{N_p} \frac{Q_i}{B_i} t_{ij} \leq H \qquad j = 1, \ldots, M \tag{21}$$

Replacing the second constraint in NLP1 by the inequalities in (21), leads to the nonlinear program NLP2 for the optimal design of multiproduct batch plants with UIS policy:

$$\min \sum_{j=1}^{M} a_j V_j^{\beta_j} \tag{NLP2}$$

s.t.

$$V_j \geq S_{ij} B_i$$

$$\sum_{i=1}^{N_p} \frac{Q_i \, t_{ij}}{B_i} \leq H \qquad j = 1, \ldots, M$$

$$V_j \geq 0 \quad j = 1, \ldots, M \; ; \qquad B_i \geq 0 \quad i = 1, \ldots, N_p$$

Note that this formulation requires M horizon constraints as opposed to the single horizon constraint in NLP1. In all, NLP2 requires $[M*N_p+M]$ constraints compared to $[M*N_p+1]$ in NLP1. The number of variables in NLP2 and NLP1 is the same since only V_j and B_i are involved. Also note that in NLP2 the investment cost of the intermediate storage is neglected.

For ZW policy the scheduling information can be embedded at the design stage as follows. Consider the assignment constraint (13),

$$\sum_{k=1}^{N} y_{ik} = 1 \qquad i = 1, \ldots, N$$

where there will be N equations, one for each of the product-batches i'. Since n_i is the number of batches for each product i, and defining S_i as the set of product-batches belonging to product i then by adding the above equations gives.

$$\sum_{i' \in S_i} \sum_{k'=1}^{N} YC_{i'k'} = n_i \qquad i = 1,\dots,N_p \tag{22}$$

$$\sum_{k'=1}^{N} \sum_{i' \in S_i} YC_{i'k'} = n_i \qquad i = 1,\dots,N_p \tag{23}$$

which leads to

$$\sum_{k'}^{N} NPRS_{ik'} = n_i \qquad i = 1,\dots,N_p \tag{24}$$

where $NPRS_{ik'}$ is the number of times that the batches of product i and product-batch k' will occur in pairs, during production run over the horizon period. By regrouping in (24) the domain of k' ($k' = 1,\dots,N$) into various products ($k=1,\dots,N_p$) leads to,

$$\sum_{k=1}^{N_p} \left[\sum_{k' \in S_k} NPRS_{ik'} \right] = n_i \qquad i = 1,\dots,N_p \tag{25}$$

$$\sum_{k=1}^{N_p} NPRS_{ik} = n_i \qquad i = 1,\dots,N_p \tag{26}$$

Similar manipulation of the assignment constraints (14) leads to:

$$\sum_{i=1}^{N_p} NPRS_{ik} = n_k \qquad k = 1,\dots,N_p \tag{27}$$

where in both (26) and (27) $NPRS_{ik}$ represents the number of times the batches of products i and k occur in pairs.

From the cycle time constraint in (15) for ZW policy, and by defining the total cycle over the time horizon H for each stage j, we get

$$CT_j^H = \sum_{i=1}^{N} \left[t_{ij} + \sum_{k=1}^{N} SL_{ikl} \; YC_{ik} \right] \qquad j = 1,\ldots,M \tag{28}$$

From (18), (24) and (27), equation (28) can be expressed as

$$CT_j^H = \sum_{i=1}^{N_p} \left[n_i \; t_{ij} + \sum_{k=1}^{N_p} SL_{ikj} \; NPRS_{ik} \right] \qquad j = 1,\ldots,M \tag{29}$$

Since this cycle time must be smaller than the horizon time H for each stage, the constraints that apply are

$$\sum_{i=1}^{N_p} \left[n_i \; t_{ij} + \sum_{k=1}^{N_p} SL_{ikj} \; NPRS_{ik} \right] \leq H \qquad j = 1,\ldots,M \tag{30}$$

Thus, replacing the horizon constraint in NLP1 by constraints (19), (26), (27) and (30) leads to the nonlinear program NLP3 for the optimal design of multiproduct batch plants with ZW policy:

$$\min \sum_{j=1}^{M} a_j \, V_j^{\beta_j} \tag{NLP3}$$

s.t.

$$V_j \geq S_{ij} \; B_i \qquad j = 1,\ldots,M \; ; \; i = 1,\ldots,N_p$$

$$n_i = \frac{Q_i}{B_i} \qquad i = 1,\ldots,N_p$$

$$\sum_{k=1}^{N_p} \text{[illegible]} \qquad i = 1,\ldots,N_p$$

$$\sum_{i=1}^{N_p} NPRS_{ik} = n_k \qquad k = 1,\ldots,N_p$$

$$\sum_{i=1}^{N_p} \left[n_i \; t_{ij} + \sum_{k=1}^{N_p} SL_{ikj} \; NPRS_{ik} \right] \leq H \qquad j = 1,\ldots,M$$

$V_j \geq 0 \quad j=1,\ldots,M \qquad n_{s_{A*}} \cdot B_j \geq 0 \quad i=1,\ldots,N_p$

$NPRS_{ik} \geq 0 \quad i,k=1,\ldots,N_p$

Note that when compared to the model NLP1 for single product campaign, the formulation NLP3 is somewhat larger as it involves N_p*M+3N_p+M constraints. Also, there are an additional $N_p(N_p)+N_p = N_p^2+N_p$ variables [NPRS^ and n^]. However, note that except for (19) all the additional constraints are linear.

Remarks

The formulations NLP1, NLP2 and NLP3 presented in the previous section exhibit a unique local optimum solution. The proof for NLP1 can be obtained by transforming the problem to a geometric programming problem [e.g. see Grossmann and Sargent, 1979]. The proof for NLP2 is identical. For NLP3 an outline of a proof is given in Appendix IV.

The formulations NLP2 and NLP3 have the important feature of considering for the optimal design mixed product campaigns with the UIS and the ZW scheduling policies, respectively. Since these limiting policies increase the utilization of the equipment, formulations NLP2 and NLP3 will produce in general designs that exhibit smaller equipment sizes than the formulation NLP1 for single product campaigns. It should be noted that from the solutions of NLP2 and NLP3, one can derive with the predicted number of batches for each product the detailed schedules. In simple cases this can be done by inspection, but generally through the use of formulations MILP3 and MILP4. However, formulations NLP2 and NLP3 can in principle produce schedules which change continuously over the specified time horizon H (e.g. 1 year). This follows from equations (20) and (29) where the cycle times for each stage were defined for the total number of product-batches that are to be produced over the hori2on time H.

In the case where the formulations predict number of batches n_i that are in integer ratios the above difficulty can be easily circumvented so as to produce schedules that involve cycles of relatively short duration that are repeated periodically. For example, assume that 100 batches are predicted for product A, 200 for product B and 150 for product C. Then, the simplest alternative to deriving a detailed schedule would be to consider cycles consisting of 2 batches of product A, 4 of B and 3 of C. Alternatively, one might consider cycles consisting of 4 batches of A, 8 of B and 6 of C. As shown in Appendix V the cycle time of the shorter cycles multiplied by the number of cycles N_c is equal to the total cycle time CT^H over the horizon time H. Hence, neglecting for the effects of switchover times, the alternative schedules are equivalent.

In a number of instances, however, formulations NLP2 and NLP3 may predict number of batches n_i that are not in integer ratios, which would then imply that the resulting schedule would have to change continuously over the time horizon H. In order to objtain more reasonable schedules, one can resolve NLP2 and NLP3 by rounding-off the ratio of number of batches. This can be accomplished as follows:

Denote by q the product with the fewest number of batches; that is,

$$n_{if} \; £ \; n_{4.} \qquad ;c=1,...,/V_r \;, \; I \; jl \; q \qquad (31)$$

If the total number of batches of product *q* to be produced in the horizon time H is n_Q, then the ratio of number of batches that is to be rounded to a rational number will be given by

$$R_{Jn} \; \bullet \; — \qquad i=\backslash,...,N_p \; I \; J \; q \qquad (32)$$

In this way by resolving NLP2 or NLP3 with constraint (32), schedules involving shorter cycles with integer number of batches can be obtained by considering at each cycle R^*fs£ batches for product *i«* where N£ and R^*N£ are integer numbers. Also the number of cycles will be given by N_c * nJfs£ .

As an example suppose that either NLP2 or NLP3 predict 124 batches of A, 186 batches of B and 380 batches of C. Here product *q* (the product having least number of batches) corresponds to product A. Then a suitable choice of the ratios in (32) would be $R_{BA}{}^{s}$1-5, $R_{CA}{}^{C}$3. By resolving the NLP problem with these constraints, shorter cycles containing each only 11 batches (2 of A, 3 of B and 6 of C) can be obtained by specifying N_A^C = 2. Since the simpler schedule will require increased equipment sizes, the designer would need to establish the trade-off between simplicity in the scheduling and investment cost of the units. This point will be illustrated in example 4.

Examples for Design and Scheduling

Example 3 :

This design problem consists of products A and B that are to be produced in a batch plant with 3 stages. Data on the size factors, processing times and economics are given in Table 5. This problem was solved with formulation NLP1 (single product campaign), NLP2 (UIS) and NLP3 (ZW) with MINOS 5 [Murtagh and Saunders, 1985] through the computer code GAMS [Meeraus and Brooke, 1985]. NLP1 required 5 variables and 7 constraints, NLP2, 5 and 9, NLP3, 11 and 15. The computer times were 1.03, 1.53 and 3.59 sec. of CPU time respectively on an IBM 3083.

As seen in Table 6, the optimal design with UIS involves the lowest investment cost, $30,185.60, which is 21% lower than the one with single product campaigns. If storage vessels are assumed to cost one fifth the cost of manufacturing units, this difference reduces to 14%. As for the design with ZW the investment cost of $35,973.50 is 7% lower than for the case of single product campaigns. Note from Table 6 that these reductions are due to the fact that with UIS and ZW the equipment sizes are significantly smaller and the number of batches is larger. Also note from Table 6 that the ratio of the numbers of batches of A and B is one for both UIS and ZW policies. From the solutions to NLP2 and NLP3 it is easy to derive the schedules which involve alternating one batch of A and one batch of B as shown in Fig. 6.

Example 4 :

This design problem consists of 6 products A, B, C, D, E and F in a plant involving 4 production stages. The data are given in Table 7. NLP1 required 10 variables and 25 constraints, NLP2, 10 and 28 , NLP3, 52 and 46. The computer times required were 2.335, 3.905 and 7.485 sec. of CPU time respectively on an IBM 3083.

As seen in Table 8, when no roundoff constraints are imposed on the ratios for the number of batches, the design with UIS has only an investment cost of $159,000 (neglecting intermediate storage). The design with ZW has a cost of $183,809 and the one with single product campaigns costs $206,298. However, the number of batches in UIS and ZW exhibit noninteger ratios which implies that the corresponding schedule would change continuously over the time horizon of 6000 hrs.

In order to obtain more reasonable schedules, the round-off constraints (32) can be imposed. For instance, in the case of ZW policy where 248, 231, 264, 248, 99 and 66 batches were obtained (see Table 8), the computed ratios are as follows:

$$\frac{n_A}{n_F} = 3.76; \quad \frac{n_B}{n_F} = 3.50; \quad \frac{n_C}{n_F} = 4.00; \quad \frac{n_D}{n_F} = 3.76; \quad \frac{n_E}{n_F} = 1.50$$

It should be recognized that the *pcuUeJtn oi tht AchidudUng i* aiitJiaxJUiy dtptndzvit on thZAZ hoJULoh.* Thus, in order to minimize the penalty in capital cost incurred by the rounding off procedure, it is important to change the above ratios by the least extent possible. The selected ratios in (32) for the first level of round-off were as follows,

"*, ' 4' «» m 3-5' "CF " 4' "O, - 4' "• - ™

From the solution of NLP3 with these ratios and by setting $N^{C} = 2$, results in a design with a schedule that consists of 32 .identical cycles, each involving the processing of 36 batches (8 of A, 7 of B, 8 of C, 8 of D, 3 of E and 2 of F). In order to obtain a schedule that will repeat with higher frequency, the ratios have to be rounded-off further. For the next level of round-off the selected ratios were

"» ' 4' *., - 4' "* * 4' *a, ' 4' "„ ' 2

From the solution of NLP3 and by setting $N^{C} = 1$, 61 cycles are obtained each involving 19 batches (4 each of A, B, C and D, 2 of E and 1 of F). The capital cost for the design resulting from first round-off is $185,424 (savings of 10.1% as compared to single product campaigns), which increases to $190,843 (savings of 7.5%) after the second round-off.

The detailed schedules for the above two designs (see Fig. 7) were derived using MILP4 presented earlier on in this paper. The design after first rounding leads to a schedule that consists of 32 cycles each containing 36 product-batches. The cycle time for **the** optimal schedule was 181 hours. The scheduling problem MILP4, consisting of 1260 integer variables and 1 continuous variable, gave integer solution when solved as an LP on an IBM 3083 using MPSX through GAMS. The CPU time required was 29.49 sec. for generation of the model and 42.25 sec. for execution and writing the output in a file. The design after second rounding leads to a schedule consisting of 61 identical cycles each containing 19 product-batches. The optimal cycle time was 105 hours. The scheduling problem MILP4, consisting of 342 integer variables and one continuous variable, needed 7.91 sec. for generating the model and 11.93 sec. for execution and reporting the solution.

For the UIS policy the following ratios were selected:

*« " 4, *„ m 3,5, "cr - 6, *or " 4, "* = 1,5

With Np = 2, the resulting design involves a schedule with 37 cycles each involving 40 batches with a cycle time of 161 hrs. The increase in the capital cost for having this simpler schedule is of only $3,066. Clearly the final selection from the designs in Table 8 will depend on the cost of intermediate storage and the preference given to simple schedule over lower capital investments.

Conclusions

This paper has presented new NLP formulations for the optimal design of multiproduct batch plants that account for the UIS and ZW scheduling policies in mixed product campaigns. These formulations rely on the use of simplified constraints that **were** derived from effective scheduling models that minimize the cycle time. The scheduling model for ZW policy has the interesting feature that it can be formulated as a 0-1 minimax problem which exhibits 0-1 solutions for the cases where one unit limits the cycle time. Otherwise it has zero gap which implies that it can be solved as an LP problem in most cases.

Two numerical examples have been presented for scheduling and two examples for the optimal design problem. The former have shown that scheduling solutions that minimize cycle time can be obtained very efficiently. The latter have shown that substantial economic savings can be obtained in the investment cost of multiproduct batch plants by anticipating efficient scheduling policies.

Acknowledgments

The authors would like to acknowledge financial support from the National Science Foundation under grant CPE-8351237 and for partial support from the Engineering Design Research Center at Carnegie Mellon University.

Appendix.I: Determination of stack times for **MILP4**

Formulation MILP4 requires the determination of the slacks of all stages (j) for all possible combinations of pairs of the N batches of N_p products. The total number of permutations of product-batches that is possible when two are taken at a time is given by, $N^2 - N$. For example, consider the data given in Table 1. Here the total number of combinations of product-batches is 6 (N=3). They are,

$$\begin{array}{c} \text{A-B} \ , \ \text{A-C} \\ \text{B-A} \ , \ \text{B-C} \\ \text{C-A} \ , \ \text{C-B} \end{array} \qquad \text{(A1)}$$

In order to determine the slacks SL_{ikj} the following simple procedure can be applied for each pair of product batches *i,k.*

1. Set the final times θ^F_{ij} for batch *i* in the M stages (j = 1,.....,M) by the equations:

$$\theta^F_{ij} = \sum_{\ell=1}^{j} t_{i\ell} \qquad j = 1,\ldots,M \qquad \text{(A2)}$$

2. Set the initial times θ^I_{kj} for batch k in the M stages (j=1,...,M) by the equations:

$$\theta^I_{k1} = \theta^F_{i1}; \quad \theta^I_{k,j+1} = \theta^F_{i1} + \sum_{\ell=1}^{j} t_{k\ell} \qquad j=1,\ldots,M-1 \qquad \text{(A3)}$$

3. a) Calculate the differences between initial and final times:

$$d_{ikj} = \theta^I_{kj} - \theta^F_{ij} \qquad j = 1,\ldots,M \qquad \text{(A4)}$$

b) Set the time violation $\bar{d}_{ik}$ to

$$\bar{d}_{ik} = \min_{j=1,\ldots,M} \{ d_{ikj} \} \qquad \text{(A5)}$$

c) Set the slacks SL_{ikj} to

$$SL_{ikj} = -\bar{d}_{ik} + d_{ikj}$$

It can be easily verified that for the example in Table 1 the above procedure yields the slacks shown in Table 4.

Appendix II: On the LP Relaxation of problem MILP4

Problem MILP4 for minimizing the cycle time with ZW policy corresponds to the following problem (see equations (12M16)):

$$\min \quad CT \qquad \text{(MILPA)}$$

s.t.

$$\sum_{i=1}^{N}\sum_{k=1}^{N} SL_{ikj}\, YC_{ik} + \sum_{i=1}^{N} t_{ij} - CT \le 0 \qquad j = 1 \ldots M$$

$$\sum_{k=1}^{N} YC_{ik} = 1 \qquad i = 1,\ldots N$$

$$\sum_{i=1}^{N} YC_{ik} = 1 \qquad k = 1,\ldots N$$

$$YC_{ik} = 0,1 \qquad i = 1,\ldots N; \quad k = 1,\ldots N \quad i \ne k; \quad CT \ge 0$$

The following proposition establishes the relation of above MILP with its LP relaxation.

<u>**PROPOSITION :**</u> **The LP relaxation of problem (MILP4) exhibits :**

(a) 0-1 solutions for the case of one stage that limits the cycle time.

(b) Zero gap for the case where 2 or more stages limit the cycle time.

<u>**Proof**</u>

(a) Consider the case when the optimal solution of MILP4 is defined by a single stage; say the element j^* of the inequalities that define the cycle time CT. MILP4 then reduces to,

$$\min \sum_{i=1}^{N}\sum_{k=1}^{N} SL_{ikj^*}\, YC_{ik} + \sum_{i=1}^{N} t_{ij^*} \qquad (B1)$$

s.t.

$$\sum_{k=1}^{N} YC_{ik} = 1 \qquad i = 1,\dots,N$$

$$\sum_{i=1}^{N} YC_{ik} = 1 \qquad k = 1,\dots,N$$

$$YC_{ik} = 0,1 \qquad i=1,\dots,N \; ; \; k=1,\dots,N, \quad i \neq k$$

Since (B1) corresponds to an assignment problem, it is well known [see Garfinkel and Nemhauser, 1972] that due to the unimodularity of its constraints, its LP relaxation yields 0-1 values for the variables YC_{ik} . Furthermore, since j^* has the largest cycle time it follows that for

$$0 \le \alpha_j \le 1, \; j=1,\dots,M, \; j \neq j^* , \quad 0 \le \alpha_{j^*} < 1, \quad \sum_{j=1}^{M} \alpha_j = 1$$

$$\sum\sum_{(i,k)\in I'} SL_{ikj^*} + \sum_{i=1}^{N} t_{ij^*} > \sum_{j=1}^{M} \alpha_j \left[\sum\sum_{(i,k)\in I'} SL_{ikj} + \sum_{i=1}^{N} t_{ij} \right] \tag{52}$$

where $I' = \{(i,k) \mid YC_{ik} = 1$ from the solution of (B1) $\}$. That is, the cycle time of stage j^* is strictly greater than a linear combination of cycle times of all the stages with $\alpha_j < 1$.

(b) Consider the case when the optimal solution of MILP4 is defined by more than one stage; that is by the inequalities $j \in J_A$, where $|J_A| \ge 2$.

Problem MILP4 then reduces to:

$$\min \; CT \tag{B3}$$

s.t.

$$\sum_{i=1}^{N}\sum_{k=1}^{N} SL_{ikj}\, YC_{ik} + \sum_{i=1}^{N} t_{ij} - CT = 0 \qquad j \in J_A$$

$$\sum_{k=1}^{N} YC_{ik} = 1 \qquad i = 1,\dots N$$

$$\sum_{i=1}^{N} YC_{ik} = 1 \qquad k = 1,\dots N$$

$$YC_{ik} = 0,1 \qquad i = 1,\dots N \; ; \; k = 1,\dots N, \quad i \neq k \; ; \quad CT \geq 0$$

Incorporating the complicating constraints on CT in the objective function through the lagrangian L, and relaxing the 0-1 constraints yields,

$$\min L = \sum_{j \in J_A} \lambda_j \left[\sum_{i=1}^{N} \sum_{k=1}^{N} SL_{ikj} YC_{ik} + \sum_{i=1}^{N} t_{ij} \right]$$

s.t.

$$\sum_{k=1}^{N} YC_{ik} = 1 \qquad i = 1,\dots N \qquad (B4)$$

$$\sum_{i=1}^{N} YC_{ik} = 1 \qquad k = 1,\dots N$$

$$0 \leq YC_{ik} \leq 1, \qquad i = 1,\dots N \; ; \; k = 1,\dots N, \quad i \neq k$$

where λ_j are the non-negative lagrange multipliers for the constraints $j \in J_A$ in (B3). These multipliers satisfy the equation.

$$\sum_{j \in J_A} \lambda_j = 1$$

which follows from the stationary condition of the lagrangian in (B3) with respect to CT.

The LP problem (B4) is also an assignment problem, and hence its solution yields 0-1 values for the variables YC_{ik}. The optimal lagrangian in (B4) will then be given by

$$L^* = \sum_{j \in J_A} \lambda_j^* \left[\sum_{(i,k)} \sum_{\in I''} SL_{ikj} \; YC_{ik} + \sum_{i=1}^{N} t_{ij} \right] \qquad (B5)$$

where $I'' = \{(i,k) \mid YC_{ik} = 1$ from the solution of (B4)}, and λ_j^* are the optimal lagrange multipliers in the LP relaxation of (B3).

It remains to be proved that there is no dual gap between L^* and the solution of MILP4. If there is a gap, there will exist at least one stage $j^* \in J_A$, for which

$$\sum_{j \in J_A} \lambda_j \left[\sum_{(i,k)} \sum_{\in I''} SL_{ikj} + \sum_{i=1}^{N} t_{ij} \right] < \sum_{(i,k)} \sum_{\in I''} SL_{ikj^*} + \sum_{i=1}^{N} t_{ij^*} \qquad (B6)$$

But from (B2) this would then imply that only one stage j^* defines the active constraint for the cycle time in MILP4, which contradicts the assumption that $|J_A| \geq 2$. Therefore, for $|J_A| \geq 2$ the LP relaxation of MILP4 has no dual gap. (QED)

It should be noted from the above proposition that 0-1 solutions from the relaxed LP of MILP4 will always be obtained if there is only one stage that limits the cycle time. For the case of two or more stages, the dual gap will be zero. Since there is no unimodularity of the constraints in MILP4 in this case, MILP4 may require a branch and bound search for the 0-1 solutions but with zero gap. Most cases in practice however, will yield 0-1 solutions when solved as an LP.

Appendix III: Solution of Minimum Makespan with Minimum Cycle Time

Problem MILP4 will determine a sequence of N batch-products, $S=\{P_1, P_2, \ldots, P_N\}$, that minimizes the cycle time. The alternative sequences $S^\ell = \{P_\ell, P_{\ell+1}, \ldots, P_N, P_1, P_2, \ldots, P_{\ell-1}\}$; $\ell = 1, \ldots, N$, $\ell \neq 1$, will also exhibit, the same cycle time since the same ordering of batches is maintained for the cycle. In order to schedule and select among the sequences S^ℓ, $\ell=1, \ldots, N$, the one that minimizes the makespan the following formulation can be applied:

Let the $s_{\ell k}$ be the elements of the ordered sequence ,

$$S^{\ell}=\{s_{\ell_1},s_{\ell_2},...,s_{\ell_N}\}=\{P_{\ell}, P_{\ell+1},... P_U, P, P_2, P_{\ell-1}\} \quad \ell=1,..,N \qquad (C1)$$

and define the binary variables Z^k,

$$Z^k = \begin{cases} 1 & \text{if sequence } S^k \text{ is selected} \\ 0 & \text{otherwise} \end{cases} \qquad k=1,....,N \qquad (C2)$$

Then, from equations (3M18), the minimization of the makespan leads to MILP problem:

$$\min \ T_t$$

s.t.

$$TF_{j\ell} = TI_{j\ell} + \sum_{k=1}^{N} Z^k \, t_{(s_{\ell k}),j} \qquad \ell=1,...,N \ ; \quad j=1,...,M$$

$$TF_{j\ell} \leq TI_{j,\ell+1} \qquad \ell=1,...,N \ ; j = 1,...,M$$

$$UIS: \quad TF_{j\ell} \leq TI_{j+1,\ell} \qquad \ell=1,...,N \ ; \ j=1,...,M-1$$

$$ZW: \quad TF_{j\ell} = TI_{j+1,\ell} \qquad \ell=1,....,N \ ; \ j=1,...,M-1$$

$$CT_j = TF_{jN} - TF_{j1} \qquad j=1,...,M$$

$$T_t \geq (TI_{j1}-TI_{11}) + N_c CT_j + (TF_{MN}-TF_{jN}) \qquad j=1,....,M$$

$$\sum^{N} Z^k = 1 \ . \quad Z^k = 0,1 \quad k=1....N$$

Note that the above MILP involves only N 0-1 binary variables. Solution of this MILP will then select among the schedules with minimum cycle time the one that leads to minimum makespan.

Appendix IV: On the Uniqueness of the Solution of NLP3

Only an outline of the proof will be presented here.

First consider the valid relaxation of (19)

$$"I. \quad * \quad \overline{B}_i \qquad \wedge\ 1 \ldots N_r \tag{D1}$$

By defining b^ = ln[B^], the above reduces to

$$Q\wedge \quad exp\backslash.-b\pounds \quad - \quad n\wedge \quad \le \quad 0 \qquad *=1,\ldots,/V_p \tag{02}$$

while the constraint on volumes with $v_j = \ln[V_j]$ leads to

$$Sj: \ exptbj-vjl \quad * \quad 1 \qquad i=1 \ldots M \ ; \ JL-\backslash \ldots N_B \tag{D3}$$

By substituting (19) into (30) and applying the transformation on B_i, leads to

$$\sum_{i=1} \left\{ Q_i exp[-b_i] \; - \sum_{k}^{N_p} SL_{ikj} NPRS_{ik} \right\} \le H \qquad i=1,\ldots M \tag{DA}$$

Finally, the objective can be expressed as

$$\min \sum_{j=1}^{M} a_j \ \exp[\beta_j - V_j] \tag{D5}$$

Thus, since (D5) involves a convex objective function subject to a set of convex constraints [<D2), (D3), (26), (27), (D4)] NLP3 has a unique local optimal solution.

Appendix V: On the relation of cycle times for schedules involving fewer number of batches

<u>Proposition:</u> Let CT^H be the cycle time of a schedule involving n^ batches of each product *JL* Also let CT be the cycle time of a schedule involving N_c shorter cycles with $R_{i,q}$ N^ batches of each product *i* (see remarks section for notation), where R_{Jr} ISE £ n :. Then CT^H = N CT for the UIS and ZW policy.

<u>Proof:</u>

a) Unlimited Intermediate Storage Policy:

We have from (18),

$$CT^H = \max_{j=1,...M} \{ \sum_{i=1}^{N} t_{ij} \} \quad (£1)$$

Since n^ = $R^\wedge_q$ n_q .

$$CT^H = \max_{j=1,...M} \{ \sum_{i=1}^{N_p} n_i \, t_{ij} \} = n_q \max_{j=1,...M} \{ \sum_{i=1}^{N_p} R_{iq} t_{ij} \} \quad (E2)$$

Also, since n_q * N_c N|

$$CT_H = N_c \max_{j=1,...M} \{ \sum_{i=1}^{N_P} N_q^c R_{iq} \, t_{ij} \} \quad (£3)$$

But N^ H_{Jn} is the number of batches cycles. Thus it follows that

$$CT'' * N_c \, CT \quad (£4)$$

b) Zero Wait Policy:

Following a similar reasoning from (15) we have

«"• ,.£?{££«*, * X',y) «>

$$CT^H = \max_{j=1,\dots,M} \left\{ \sum_{i=1}^{N_P} \sum_{k=1}^{N_P} n_{ik} \; SL_{ikj} \; + \; \sum_{i=1}^{N_P} n_i \; t_{ij} \right\} \qquad (E6)$$

$$CT^H = n_q \max_{j=1,\dots,M} \left\{ \sum_{i=1}^{N_P} \sum_{k=1}^{N_P} R_q^{ik} \; SL_{ikj} \; + \; \sum_{i=1}^{N_P} R_{iq} \; t_{ij} \right\}$$

$$CT^H = N_c \max_{j=1,\dots,M} \left\{ \sum_{i=1}^{N_P} \sum_{k=1}^{N_P} N_q^c R_q^{ik} SL_{ikj} + \sum_{i=1}^{N_P} N_q^c R_{iq} t_{ij} \right\}$$

$CT^H = N_c$ C7- (£7)

where

$$\sum_{k=1}^{N_P} n_{ik} = n_i \; ; \; \sum_{i=1}^{N_P} n_{ik} = n_k \; ; \; R_{iq} = \sum_{k=1}^{N_P} R_q^{ik} \text{ and } \; n_q$$ =^c*/Vj (58)

Q.E.D.

Thus, if the ratio among the number of batches of various products is maintained, minimizing the cycle time of a cycle containing a smaller number product-batches will result in a cycle time that is equivalent to that of the global cycle containing *cdUL* the product-batches in the larger time horizon.

References

1. Flatz, W. "Equipment Sizing for Multiproduct Plants", *Chem. Engg. 87,* 4 (1980), 71-80.

2. Garfinkel, R. S. and Nemhauser, G. L. *Integer Programming.* John Wiley and Sons, Inc., 1972.

3. Grossmann, I. E. and Sargent, R. W. H. "Optimum Design of Multipurpose Chemical Plants". *Ind. Eng. Chem. Proc. Des. Dev 18* (1979), 343-348.

4. Ku, Hong-ming and Karimi, I. A. "Scheduling in Multistage Serial Batch Processes with Finite Intermediate Storage- Part I: MILP Formulation". *AIChE Annual Meeting, Miami Nov.* (1986).

5. Ku, Hong-ming and Karimi, I. A. "Scheduling in Multistage Serial Batch Processes with Finite Intermediate Storage- Part II: Approximate Algorithms". *AIChE Annual Meeting, Miami* (1986).

6. Meeraus, A. and Brooke, T. "GAMS". *Development Research Department The World Bank* (1985).

7. Murtagh, B. A. and Saunders, M. A. "MINOS User's Manual". *Stanford University Palo Alto CA* (1985).

8. Papadimitriou, C. H. and Steiglitz, K. *Combinatorial Optimization: Algorithms and Complexity.* Prentice-Hall, 1982.

9. Rippin, D. W. T. "Simulation of Single and Multiproduct Batch Chemical Plants for Optimal Design and Operation". *Computers and Chemical Engineering 7,* 3 (1983), 137-156.

10. Sparrow, R. E., Forder, G. J. and Rippin, D. W. T. "The Choice of Equipment Sizes for Multiproduct Batch Plants: Heuristics v/s Branch and Bound". *Ind. Eng. Chem. Proc. Des. Dev 14,* 3 (1975), 197-203.

11. Yeh, N. C. C. and Reklaitis, G. V. "Synthesis and Sizing of Batch/Semicontinuous Processes". *AIChE Annual Meeting, Chicago, Illinois Nov 12* (1985).

LIST OF TABLES

Table 1 : Data for Example 1

Processing Times (Hrs.):

t_{ij}	A	B	C
Stage 1	2	4	3
Stage 2	5	1	2
Stage 3	4	2	5

Production Task:

To produce one batch each of products A, B and C in 3 cycles.

Table 2 : Data for Example 2

Processing Times (Hrs.):

'•|	A	B	C	D	E	F
Stage 1	10	15	20	14	6	13
Stage 2	20	8	7	6	11	7
Stage 3	5	12	9	15	5	17
Stage 4	30	10	5	10	15	10

Production Task:

To produce one batch each of products A, B, C, D , E and F in 5 cycles.

Table 3: Computational Results for Example 2

	Makespan Minimization		Cycle time Minimization	
	UIS	2W	UIS	ZW
Formulation	MILP1	MILP2	MILP3	MILP4
MILP Solution [Hrs]	427	505	80	97
LP Solution [Hrs]	423.4	423.4	80	97
0-1 Variables	36	36	36	30
Continuous Var.	53	53	49	1
Rows	87	87	83	18
Branches	40	23	5	0
Pivots	1167	1921	479	25
CPU time* [sec]	154	135	26	0.93

* Using LINDO on DEC-20.

Table 4: Slacks for Example 2

SLj_{kj} [Hrs.]

7——-i!^	A-B	A-C	B-A	B-C	C-A	C-B
Stage 1	4	3	0	0	0	2
Stage 2	3	2	1	3	0	4
Stage 3	0	0	4	3	0	0

Table 5: Data for Example 3

Production requirements : Q_A = 40,000 Kg. Q_B = 20,000 Kg.

Cost Coefficients: a_j = $ 250 p_j = 0.6

Horizon Time : H = 6000 Hrs.

Size Factors (Litres / Kg.) :

S_{ij}	A	B
Stage 1	2	4
Stage 2	3	6
Stage 3	4	3

Batch Processing Times (Hrs) :

t_{ij}	A	B
Stage 1	8	16
Stage 2	20	4
Stage 3	8	4

Table 6 : Optimal Design for Example 3

Sequencing Policies	Volumes (Utres)			No. of Batches		Batch Sizes (Kg.)		Capital Cost ($)
	V1	V2	V3	n_A	n_B	B_A	B_B	
Single Prod. Campaigns	480	720	960	167	167	240	120	38,499.80
Zero Wait	429	643	857	187	187	214	107	35,973.50
Unlimited Int. Storage	320	480	640	250	250	160	80	30,185.60*

* Cost of Intermediate Storage is not included.

Table 7: Data for Example 4

Q (Production Requirements [Kg.]) :

Product A	300,000
Product B	200,000
Product C	400,000
Product D	300,000
Product E	100,000
Product F	100,000

Cost Coefficients : α_j = \$ 250 β_j = 0.6

Horizon Time : H = 6000 Hrs.

Size Factors (litres/Kg.)

S_{ij}	A	B	C	D	E	F
Stage 1	2	7	1	5	1	4
Stage 2	3	3	4	5	6	1
Stage 3	2	1	3	2	3	1
Stage 4	6	2	2	6	2	4

Batch Processing Times (Hrs.) :

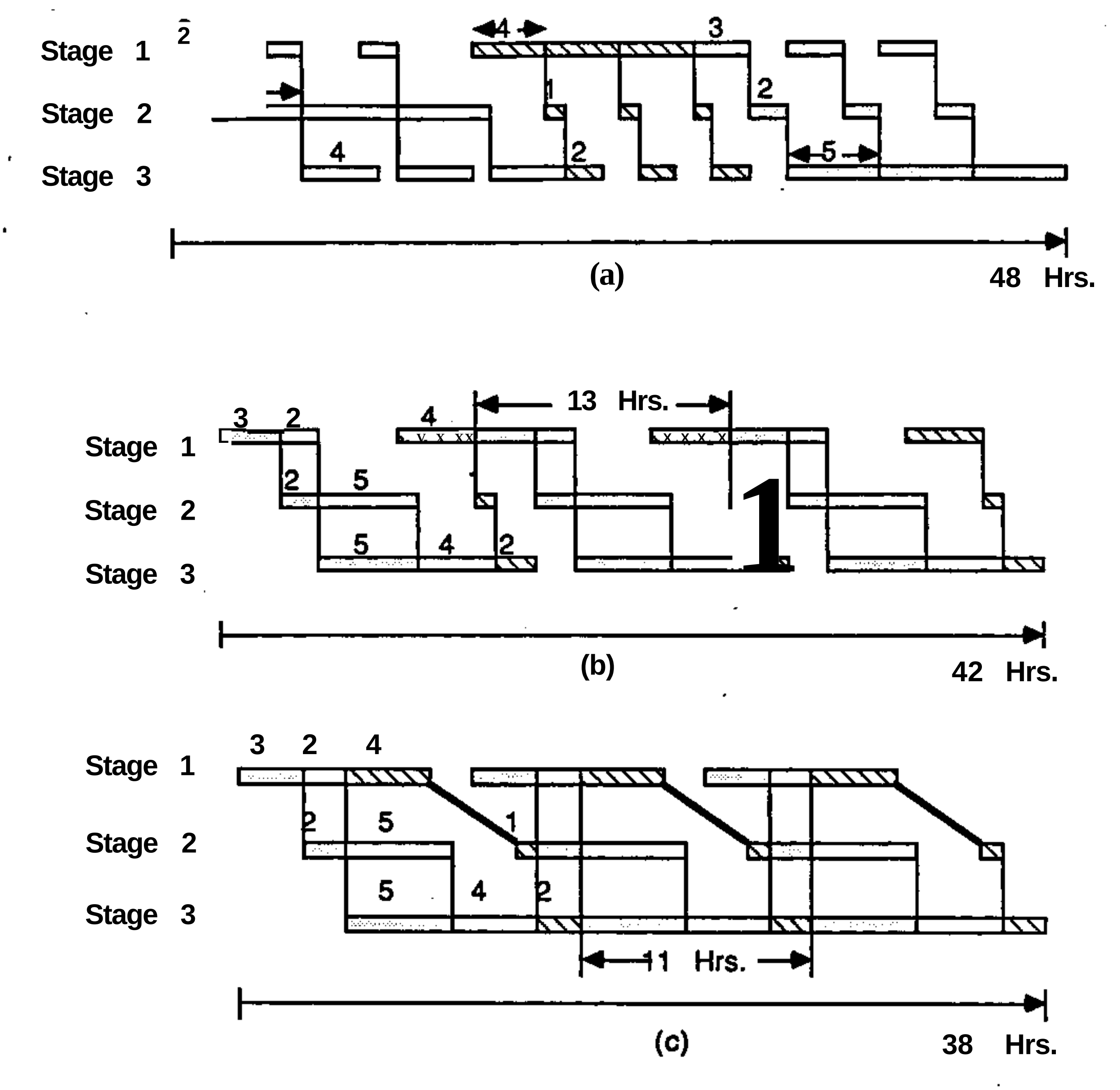
t_{ij}	A	B	C	D	E	F
Stage 1	6	1	2	8	4	3
Stage 2	2	5	7	1	1	6
Stage 3	4	3	3	5	2	2
Stage 4	1	5	7	2	2	4

Table 8: Optimal Design for Example 4

		Single Product Campaigns	Zero Wait			Unlimited Interm. Storage	
			No roundoff One Cycle / H.	1 st roundoff 32cyc!es/H.	2nd roundoff 61 nvdfts / H.	No roundoff One cvnlfi / H.	Rounded off 37 rvrles / H
V o — s (5 £)	V1	7333.33	6050	6250	6557.4	5100	5405.4
	V2	7333.33	6050	6250	6557.4	5100	5405.4
	V3	5500	4537.5	4687.5	4918	2660.87	2702.7
	V4	8800	7260	7031.25	7377	6120	6081.1
^ Sf [B — £ oo z	Prod. A	205[1466.67]	248 [1209.7]	256 [1171.9]	244 [1229.5]	294 [1020]	296 [1013.5]
	Prod. B	191[1047.62]	231 [865.9]	224 [892.9]	244 [819.7]	275 [727.3]	259 [772.2]
	ProdC	219[1833.33]	264 [1515.2]	256 [1562.5]	244 [1639.3]	451 [886.9]	444 [900.9]
	Prod. D	205[1466.67]	248 [1209.7]	256 [1171.9]	244 [1229.5]	294 [1020]	296 [1013.5]
	Prod. E	82[1222.22]	99 [1010.1]	96 [1041.7]	122 [819.7]	118 [847.5]	111 [900.9]
	Prod. F	55[1833.33]	66 [1515.2]	64 [1562.5]	61 [1639.3]	78[1282.1]	74 [1351.4]
COST($)		206,298	183,809	185,424	190,843	159,000*	162,066*
Savings over Sin. Prod. Camf.		------	10.9%	10.1 %	7.5 %	22.9 %	21.5%

* Cost of Intermediate Storage not included.

LIST OF FIGURES



IZU Product A ES Product B HH Product C Intermediate Storage

Figure 1 : Scheduling policies for Example 1 :
(a) Single Product Campaigns (b) ZW Policy (c) UIS Policy

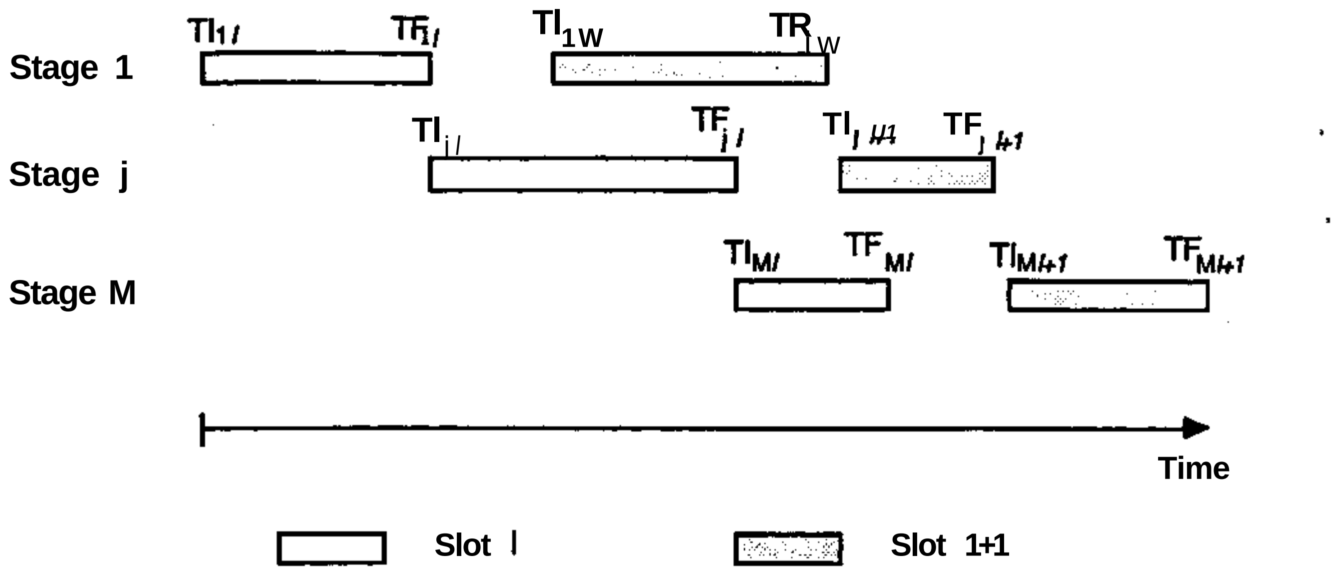


Fig. 2: Start and completion times for production slots land 1+1.

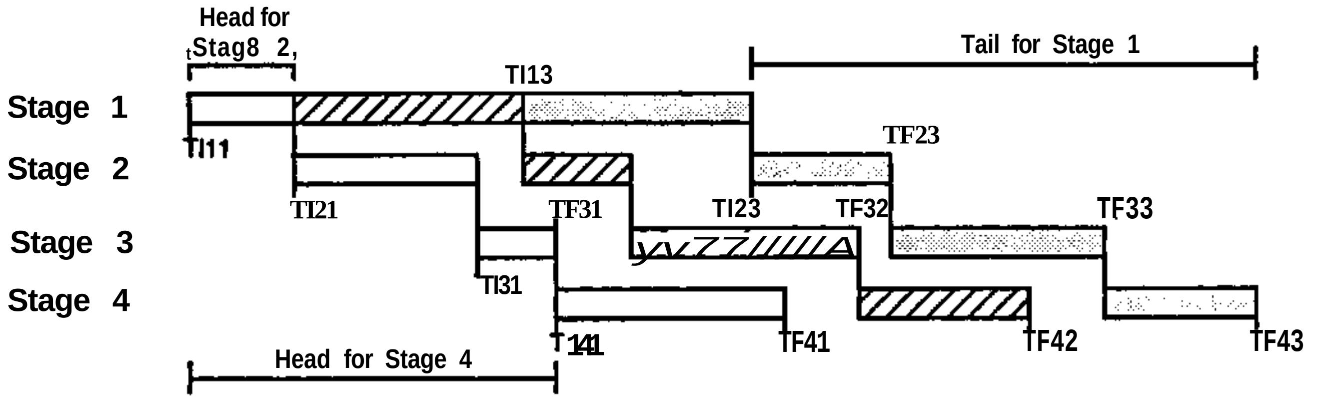


Fig. 3: Heads and tails in schedule for ZW policy.

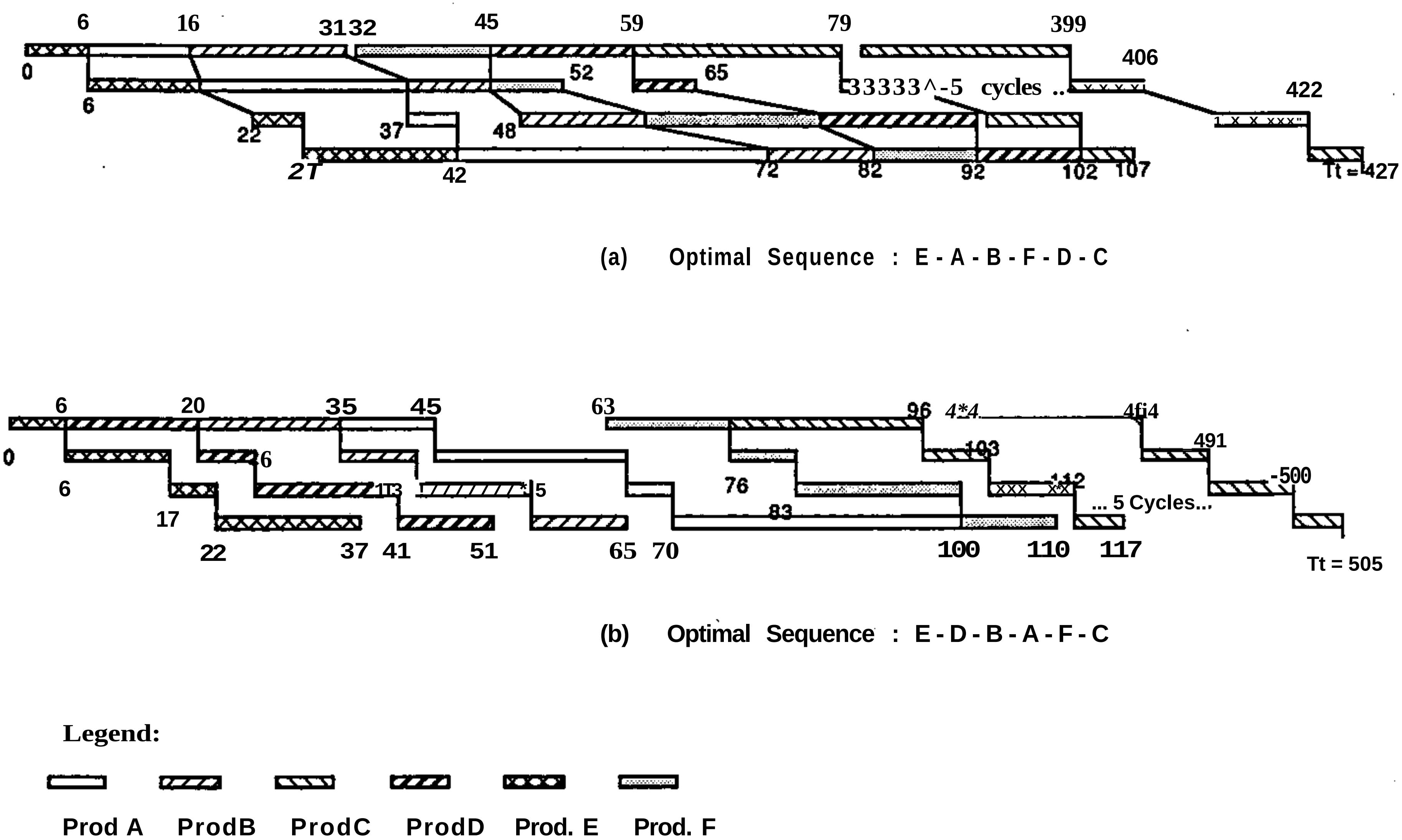


Fig. 4: Optimal Schedules for example 2;
(a) UIS Policy, (b) ZW Policy

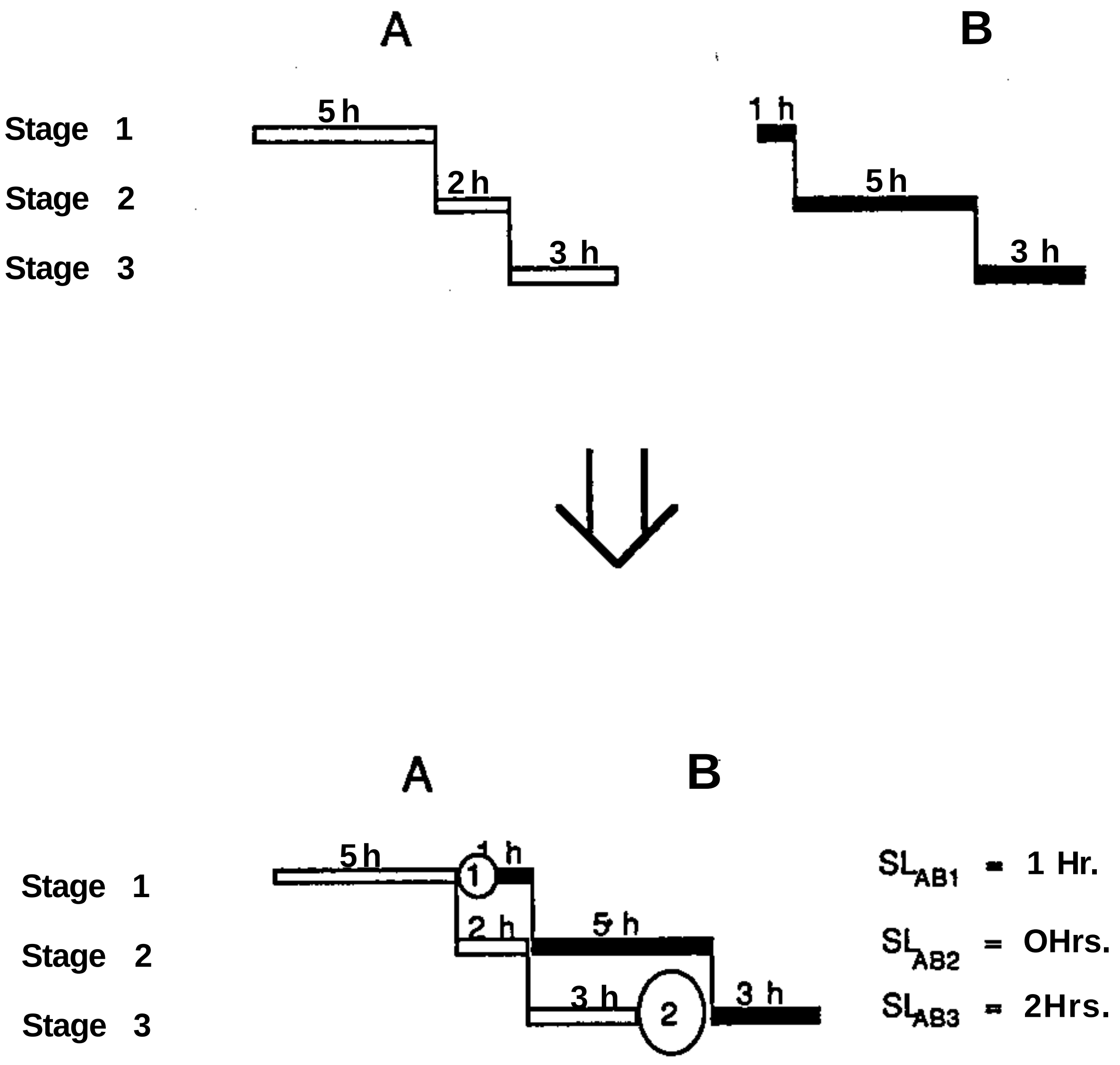


Fig. 5 : Slack Times Between Two Consecutive Batches

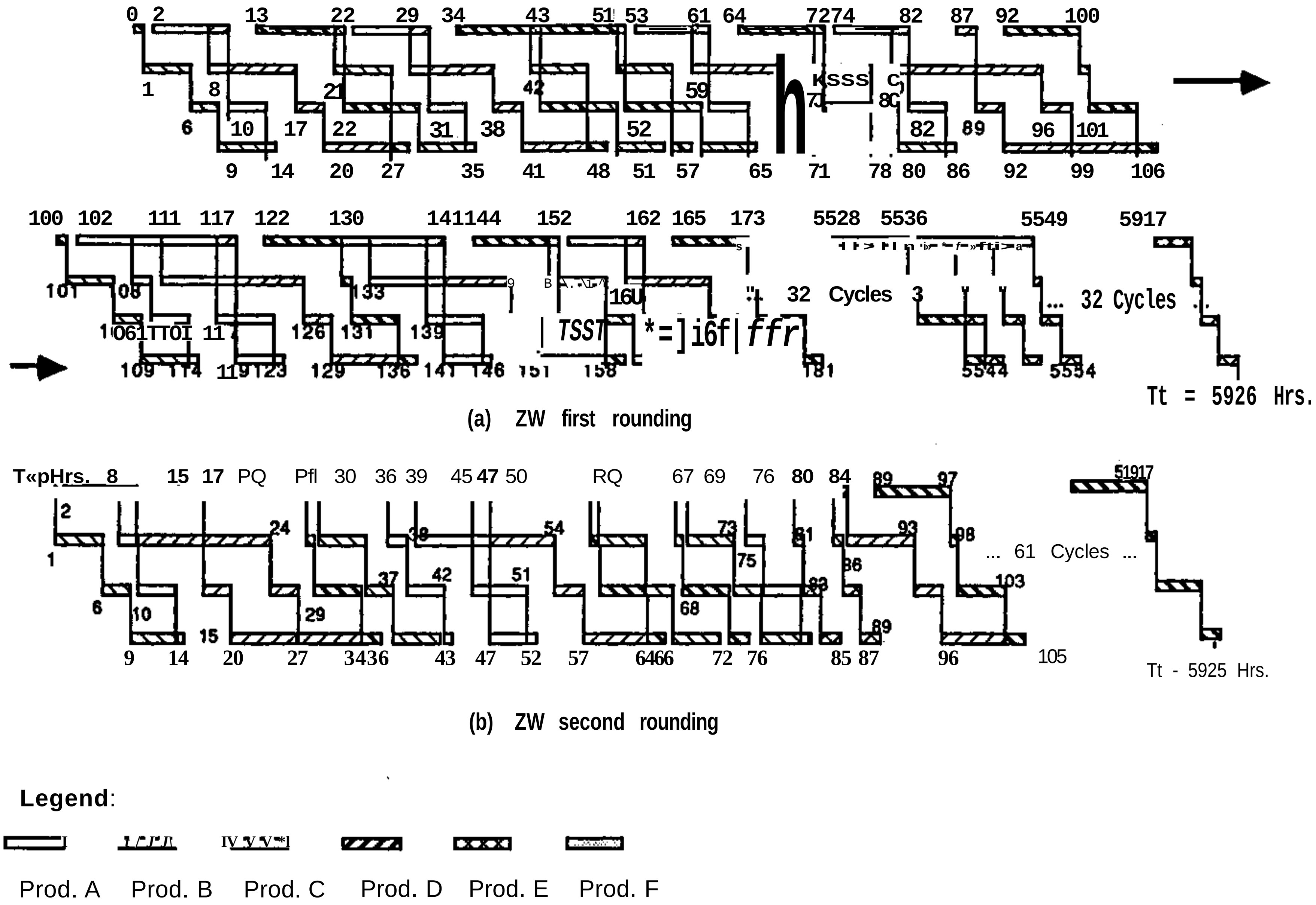


Fig. 7: Optimum Schedules for Example 4
(a) Schedule for design after first rounding
(b) Schedule for design after second rounding

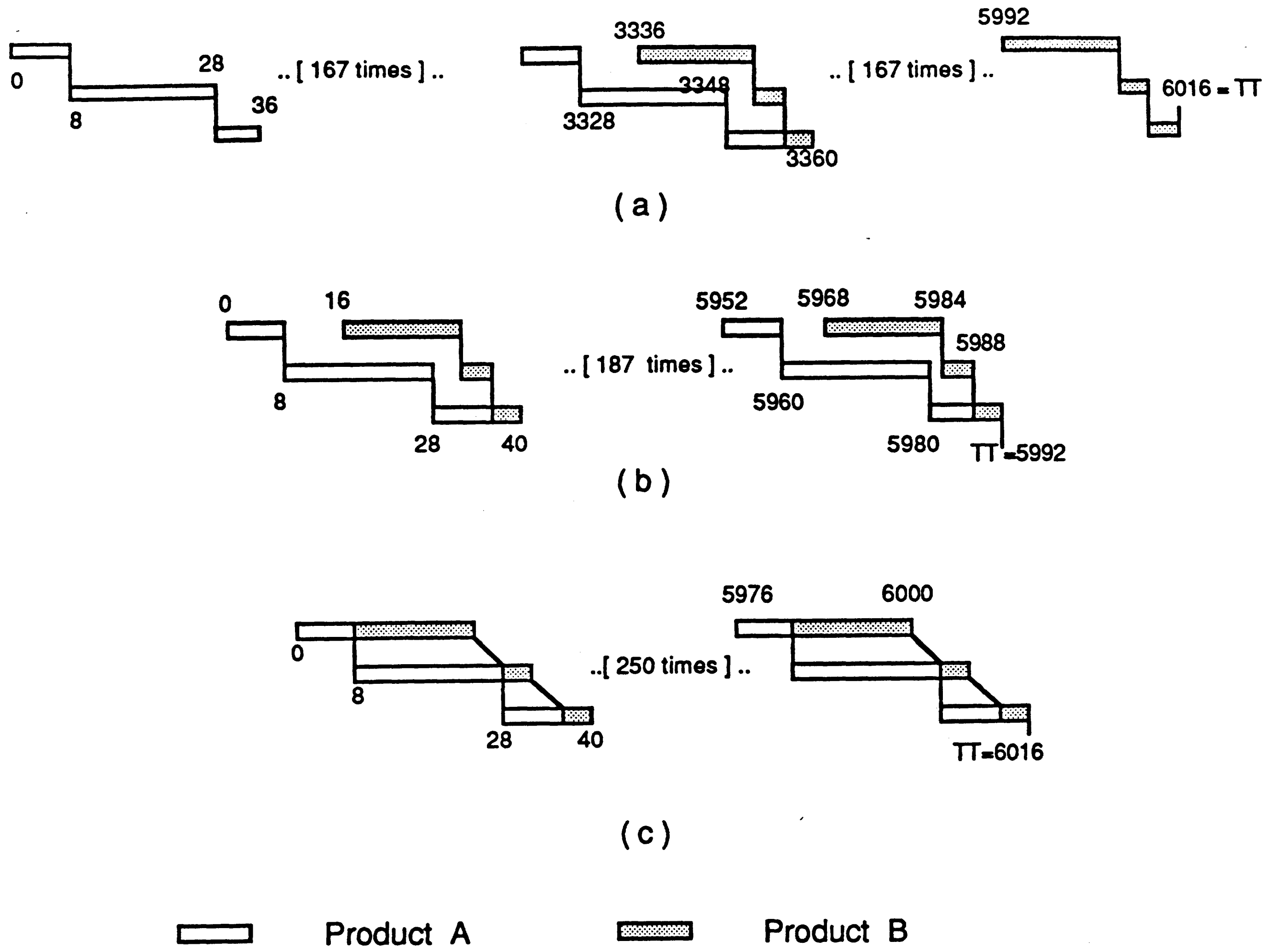


Fig. 6 Optimal Schedules for Example 3;
(a) Single Product Campaigns (b) ZW Policy (c) UIS Policy